01-03-00

Please type a plus sign (+) inside this box → [+]

PTO/SB/05 (2/98)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1 53(b))

Attorney Docket No.	042390.P6349
First Inventor or Application Identifier	Rajeev K. Nalawadi
Title	82C59 Redirection to I/O APIC
Express Mail Label No.	EM560890943US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. [X] Fee Transmittal Form *(Submit an original, and a duplicate for fee processing)*
2. [X] Specification *Total Pages* 22 *(preferred arrangement set forth below)*
   - Descriptive title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
3. [X] Drawing(s) *(35 CFR 113)* *Total Sheets* 3
4. Oath or Declaration *Total Pages*
   a. [ ] Newly executed (original copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 16 completed)* ***[Note Box 5 below]***
      i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. [ ] Microfiche Computer Program *(Appendix)*
6. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

ACCOMPANYING APPLICATION PARTS

7. [ ] Assignment Papers (cover sheet & document(s))
8. [ ] 37 CFR 3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
9. [ ] English Translation Document *(if applicable)*
10. [ ] Information Disclosure Statement (IDS)/PTO - 1449 [ ] Copies of IDS Citations
11. [ ] Preliminary Amendment
12. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
13. [ ] *Small Entity Statement(s) [ ] Statement filed in prior application, Status still proper and desired
14. [ ] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
15. [ ] Other: . . . . . .

16. **If a CONTINUING APPLICATION,** *check appropriate box, and supply the requisite information below and in a preliminary amendment:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No:_____/_____

Prior application Information: Examiner_____ *Group/Art Unit:* _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

[ ] *Customer Number of Bar Code Label* *(Insert Customer No. or Attach bare code label here)* *or* [X] *Correspondence address below*

Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP				
Address	12400 Wilshire Boulevard, Seventh Floor				
City	Los Angeles	State	California	Zip Code	90025
Country	U.S.A.	Telephone	(310) 207-3800	Fax	(310) 820-5988

Name (Print/Type)	William Thomas Babbitt, Reg. No. 39,591		
Signature	William T Babbitt	Date	12/30/99

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Attorney's Docket No. 042390.P6349
Express Mail No. EM560890943US

UNITED STATES PATENT APPLICATION

FOR

82C59 REDIRECTION TO I/O APIC

Inventor:

Rajeev K. Nalawadi

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025
(310) 207-3800

82C59 REDIRECTION TO I/O APIC

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates to computer system peripheral connections and more particularly to managing system interrupts.

Background

Peripheral Component Interconnect (PCI)-Extended Industry Standard Architecture (EISA) bridge sets provide an I/O subsystem for many computer systems. One Peripheral Component Interconnect (PCI) standard is PCI Local Bus standard version 2.2 (January 5, 1999). The chip set generally consists of two components--the PCI-EISA Bridge (PCEB) and the EISA system component (ESC). In general, the ESC implements system functions such as timer-counter, direct memory access (DMA), and interrupt control.

In one form of interrupt control utilized in certain chip set configurations of Intel Corporation of Santa Clara, California, an EISA compatible interrupt controller of the ESC incorporates the functionality of two 82C59 interrupt controllers that are cascaded providing fourteen external and two internal interrupts. The ESC also contains an Advanced

Programmable Interrupt Controller (APIC). The APIC can be used in either a uni-processor or multi-processor system. The APIC provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. In systems with multiple I/O subsystems, each system can have its own set of interrupts.

As noted, the EISA compatible interrupt controller incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded into a master interrupter controller and a slave interrupt controller. Two internal interrupts are used for internal function only and are not available at the chip periphery. One interrupt is used to cascade the two controllers together and another is used as a system timer interrupt. The remaining 14 interrupt lines are available for external system interrupts. The interrupts are programmed to utilize on the order of 2000 logic gates. Examples of uses of the 14 system interrupts include interrupts for a keyboard, hard drive, modem, etc.

While the standard EISA compatible interrupt controller is intended for use in a uni-processor system, the APIC can be used in either a uni-processor or multi-processor system. An APIC provides multi-processor interrupt management and incorporates static and dynamic symmetric interrupt distribution across all

processors. In systems with multiple I/O subsystems, each subsystem can have its own set of interrupts.

At the system level, an APIC consists of two parts--one residing in the I/O subsystem (I/O APIC) and the other in the CPU (local APIC). The ESC contains the I/O APIC unit.

The I/O APIC unit consists of a set of interrupt input signals, a 16-entry Interrupt Redirection Table, programmable registers, and a message unit for sending and receiving APIC messages over the APIC bus. I/O devices inject interrupts into the system by asserting one of the interrupt lines to the I/O APIC. The I/O APIC selects a corresponding entry in the redirection table and uses the information of that entry to format an interrupt request message. Each entry in the redirection table can be individually programmed to indicate edge/level sensitive interrupt signals, the interrupt vector and priority, the destination processor, and how the processor is selected (e.g., statically or dynamically). The information in the table is used to transmit a message to other APIC units via the APIC bus.

In addition to its compatibility with multi-processor environments, the APIC system offers the ability to handle a greater number of system interrupts with greater flexibility than the EISA compatible interrupt controller. Nevertheless, the EISA compatible interrupt controller remains a legacy

standard for interrupt control in the modern processor environment.

SUMMARY OF THE INVENTION

In one aspect, a method is disclosed. The method includes trapping initializing data of a first interrupt type to a first interrupt controller. The initializing data of the first interrupt type is re-routed to a second interrupt controller. The second interrupt controller is configured to manage interrupts of the first interrupt type.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, aspects, and advantages of the invention will become more thoroughly apparent from the following detailed description, appended claims, and accompanying drawings in which:

Figure 1 is a schematic block diagram of a multi-processor environment illustrating the interrupt control for the system.

Figure 2 is a flow chart of an embodiment of the method of the invention.

Figure 3 is a specific embodiment of the method of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In one aspect, the invention relates to a method and system to handle all system interrupts through the APIC system, including processor systems that rely for at least a portion of their interrupt control on the legacy standard of the EISA compatible interrupt controller. In this manner, the invention allows systems that rely partially on EISA compatible interrupt controllers, such as the 82C59 interrupt controller to be incorporated in a multi-processor environment. By utilizing the APIC system of interrupt control the number of dedicated gates required for interrupt control can be reduced.

Figure 1 schematically illustrates a block diagram of a multi-processor environment suitable for interrupt control according to the invention. System 100 is a multi-processor system, including processor 110, processor 210, and processor 310. Each processor contains a memory element (e.g., random access memory, read only memory, cache, etc.) System 100 also includes, in this illustration, EISA system component (ESC) 410. ESC 410 incorporates the standard EISA compatible interrupt controller and the APIC. In one embodiment, the EISA compatible interrupt controller includes two 82C59 interrupt controllers 430 arranged in a master (CNTRL-1)-slave (CNTRL-2) configuration as known in the art. In one embodiment, the two 82C59 interrupt

controllers are utilized to control the interrupt functions listed in Table 1:

Table 1: Typical Interrupt Functions

Priority	Label	Controller	Typical Interrupt Source
1	IRQ0	1	Interval Timer 1, Counter 0 OUT
2	IRQ1	1	Keyboard
3-10	IRQ2	1	Interrupt from Controller 2
3	IRQ8#	2	Real Time Clock
4	IRQ9	2	Expansion Bus Pin B04
5	IRQ10	2	Expansion Bus Pin D03
6	IRQ11	2	Expansion Bus Pin D04
7	IRQ12	2	Expansion Bus Pin D05
8	IRQ13	2	Coprocessor Error, Chaining
9	IRQ14	2	Fixed Disk Drive Controller Expansion Bus Pin D07
10	IRQ15	2	Expansion Bus Pin D06
11	IRQ3	1	Serial Port 2, Expansion Bus B25
12	IRQ4	1	Serial Port 1, Expansion Bus B24
13	IRQ5	1	Parallel Port 2, Expansion Bus B23
14	IRQ6	1	Diskette Controller, Expansion Bus B22
15	IRQ7	1	Parallel Port 1, Expansion Bus B21

The following demonstrates the interrupt sequence for a 80x86-type system.

1. One or more of the Interrupt Request (IRQ[x]) lines are raised high, setting the corresponding Interrupt Request Register (IRR) bit(s).
2. The Interrupt Controller evaluates these requests, and sends an INTR to the CPU, if appropriate.
3. The CPU acknowledges the INTR and responds with an interrupt acknowledge cycle. This cycle is translated

into a PCI bus command. This PCI command is broadcast over the PCI bus as a single cycle.

4. Upon receiving an interrupt acknowledge cycle from the CPU over the PCI, the PCI-EISA bridge (PCEB) converts the single cycle into an INTA# pulse to the ESC. The ESC uses the INTA# pulse to generate the two cycles that the internal 82C59 pair can respond to with the expected interrupt vector. The cycle conversion is performed by a functional block in the ESC Interrupt Controller Unit. The internally generated interrupt acknowledge cycle is completed as soon as possible as the PCI bus is held in wait states until the interrupt vector data is returned. Each cycle appears as an interrupt acknowledge pulse on the INTA# pin of the cascaded interrupt controllers. These two pulses are not observable at the ESC periphery.

5. Upon receiving the first internally generated interrupt acknowledge, the highest priority In-Service Register (ISR) bit is set and the corresponding IRR bit is reset. The Interrupt Controller does not drive the Data Bus during this cycle. On the trailing edge of the first cycle pulse, a slave identification code is broadcast by the master to the slave on a private, internal three bit wide bus. The slave controller uses these bits to

determine if it must respond with an interrupt vector during the second INTA# cycle.

6. Upon receiving the second internally generated interrupt acknowledge, the Interrupt Controller releases an 8 bit pointer (the interrupt vector) onto the Data Bus where it is read by the CPU.

7. This generally completes the interrupt cycle. In an automatic end-of-interrupt mode the ISR bit is reset at the end of the second interrupt acknowledge cycle pulse. Otherwise, the ISR bit remains set until an appropriate end-of-interrupt (EOI) command is issued at the end of the interrupt subroutine.

At the system level, the APIC consists of two parts -- one residing in the I/O subsystem (I/O APIC 420) and the other, local APIC 120, 220, and 320, residing in processor 110, processor 210, and processor 310, respectively. The local APIC and I/O APIC 420 communicate over dedicated APIC bus 400. The ESC's I/O APIC bus interface consists of bi-directional data signals and a clock input. Each local APIC (local APIC 120, local APIC 220, and local APIC 320) contains intelligence to determine whether or not each processor should accept interrupts broadcast on APIC bus 400. The local APIC unit also provides local pending of interrupts, nesting and masking of interrupts, and handles all interactions with its local processor (e.g.,

interrupt request (INTR), interrupt acknowledge (INTA, and EOI protocol). Each local APIC also provides inter-processor interrupts and a timer to its local processor.

As shown in **Figure 2**, the invention contemplates identifying and redirecting EISA compatible interrupt signals (block 450), masking the EISA compatible interrupt controller interrupt addresses (block 460) and re-directing the interrupt addresses to the I/O APIC (block 470). In this manner, some or all of the interrupts handled by the EISA compatible interrupt controller (e.g., the 82C59 controller(s)) are instead handled by the I/O APIC.

Figure 3 describes a method of redirecting 82C59 interrupt addresses to an I/O APIC. This embodiment takes advantage of the normal process of initializing interrupt registers on system start-up. The embodiment seeks to identify 82C59 interrupt addresses through system logic. One way this is accomplished is by configuring a system management interrupt (SMI) to trap a configuring 82C59 address (block 510). An SMI is derived from power management architecture of the ESC. In particular, the system management mode (SMM) is a function of the ESC's power management architecture that includes software that controls transitions between a Power-On state and a Fast-Off state. The Fast-Off state is generally utilized in modern processors to permit a system to operate in a low power state without being

powered down. When the system is in a Fast-Off state, the system consumes less power than the Power-On state and is not in a Power Off state. Events such as pressing a key on a keyboard or moving a mouse constitutes a Fast-On event to convert from a Fast-Off state to a Power-On state.

To invoke SMM software, the ESC generates an SMI to the processor. An SMI offers serialized execution. Once an SMI happens, the processor stops and processes the SMI.

The 82C59 generally contains four initializing address registers -- 20h, 21h, A0h, A1h. The four address registers are initialized by four initialization command words: ICW1, ICW2, ICW3, and ICW4. Before normal operation can begin, each interrupt controller in the system must be initialized. In the ESC, this is a four byte sequence. The base address for each interrupt controller is a fixed location in I/O memory space, at 0020h for CNTRL-1 and 00A0h for CNTRL-2. An I/O write to the CNTRL-1 or CNTRL-2 base address with data bit 4 equal to 1 is interpreted as ICW1. For ESC-based EISA systems, three I/O writes to "base address + 1" (021h for CNTRL-1 and 0A0h for CNTRL-2) must follow the ICW1. The first write to "base address + 1" (021h/0A0h) performs ICW2, the second write performs ICW3, and the third write performs ICW4. Initialization of the 82C59 controller thus generally occurs in a particular sequence, ICW1, ICW2, ICW3, and ICW4.

In one embodiment, an SMI is configured to recognize and trap a configuring 82C59 address, such as one of the identified initialization command words. In addition to trapping a configuring 82C59 address, the trapped address is analyzed to determine whether or not the address is the first configuring address (block 520). As noted above, 82C59 is configured in a specific sequence, therefore, prior to re-routing the initialization addresses to I/O APIC as a trap, the SMI locates the first configuring address. A look-up table containing the appropriate order of the initialization command words is provided to identify the appropriate configuring address.

Once the SMI locates the first configuring address, the SMI then locates the remaining configuring addresses (block 530). An SMI vector handler is then generated to direct the appropriate processor to find the interrupt (block 540). The SMI vector handler is associated with the I/O APIC. Once the SMI vector handler is generated, an SMI vector is passed to the appropriate processor (block 550). In response to the SMI vector, the processor re-routes configuring addresses to the I/O APIC (block 560). In this regard, the I/O APIC is then configured to handle 82C59 interrupts.

By handling all 82C59 interrupts through the APIC system, the invention supports the use of conventional chip set systems with advanced operating systems that do not support 82C59

interrupt control. The invention also supports the use of such chip set systems in multi-processor environments.

In the preceding detailed description, the invention is described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method comprising:

trapping initializing data of a first interrupt type to a first interrupt controller;

re-routing the initializing data of the first interrupt type to a second interrupt controller; and

configuring the second interrupt controller to manage interrupts of the first interrupt type.

2. The method of claim 1, wherein trapping initializing data of a first interrupt type comprises:

configuring a system management interrupt to recognize initializing data of a first interrupt type.

3. The method of claim 1, wherein initializing data of the first interrupt type comprises a plurality of command words and a first command word begins the initializing of the first interrupt controller, configuring a system management interrupt to recognize initializing data of a first interrupt type and re-route initializing data to the second interrupt controller from the first command word.

4. The method of claim 1, wherein the first interrupt controller comprises an 82C59 controller and the second interrupt controller comprises a advanced programmable interrupt controller.

5. A machine readable storage media containing executable program instructions which when executed cause a digital processing system to perform a method comprising:

trapping initializing data of a first interrupt type to a first interrupt controller;

re-routing initializing data of a first interrupt type to a second interrupt controller; and

configuring the second interrupt controller to manage interrupts of the first interrupt type.

6. The media of claim 5, wherein trapping initializing data of a first interrupt type comprises:

configuring a system management interrupt to recognize initializing data of a first interrupt type.

7. The media of claim 5, wherein initializing data of the first interrupt type comprises a plurality of command words and a first command word begins the initializing of the first

interrupt controller, configuring a system management interrupt to recognize initializing data of a first interrupt type and re-route initializing data to the second interrupt controller from the first command word.

8. The media of claim 5, wherein the first interrupt controller comprises an 82C59 controller and the second interrupt controller comprises a advanced programmable interrupt controller.

9. A system comprising:

a central processing unit (CPU);

a first bus coupled to the CPU;

a first interrupt controller, coupled to the first bus, operable to manage communication with the CPU of interrupts of a first interrupt type;

a second bus coupled to the CPU;

a second interrupt controller, coupled to the second bus and to the first interrupt controller, operable to manage communication with the CPU of interrupts of a second interrupt type; and

a memory coupled to the second interrupt controller comprising a computer-readable medium having a computer-readable

program embodied therein for directing operation of the system, the computer-readable program comprising:

instructions for managing interrupts of the first interrupt type by the second interrupt controller.

10. The system of claim 9, wherein the computer-readable program further comprises:

instructions for trapping initializing data of a first interrupt type to the first interrupt controller;

instructions for re-routing initializing data of a first interrupt type to the second interrupt controller; and

instructions for configuring the second interrupt controller to manage interrupts of the first interrupt type.

11. The system of claim 10, wherein the instructions for trapping initializing data comprise:

instructions for configuring a system management interrupt to recognize initializing data of a first interrupt type.

12. The system of claim 10, wherein initializing data of the first interrupt type comprise a plurality of command words and a first command word begins the initializing of the first interrupt controller, and the computer-readable program comprises instructions for configuring a system management

interrupt to recognize initializing data of a first interrupt type and re-route initializing data to the second interrupt controller from the first command word.

13. The system of claim 9, wherein the first interrupt controller comprises an 82C59 controller and the second interrupt controller comprises a advanced programmable interrupt controller.

14. The system of claim 13, wherein the second interrupt controller manages interrupts of the first interrupt type exclusive of the first interrupt controller.

15. A system comprising:

a central processing unit (CPU);

first means of interrupt processing for managing communication with the CPU of interrupts of a first interrupt type;

second means of interrupt processing for managing communication with the CPU of interrupts of a second interrupt type;

means for routing interrupts of the first interrupt type to the second interrupt processing means

means for managing interrupts of the first interrupt type by the second interrupt processing means exclusive of the first interrupt processing means.

ABSTRACT

In one aspect, a method is disclosed. The method includes trapping initializing data of a first interrupt type to a first interrupt controller, re-routing the initializing data of the first interrupt type to a second interrupt controller, and configuring the second interrupt controller to manage interrupt of the first interrupt type.

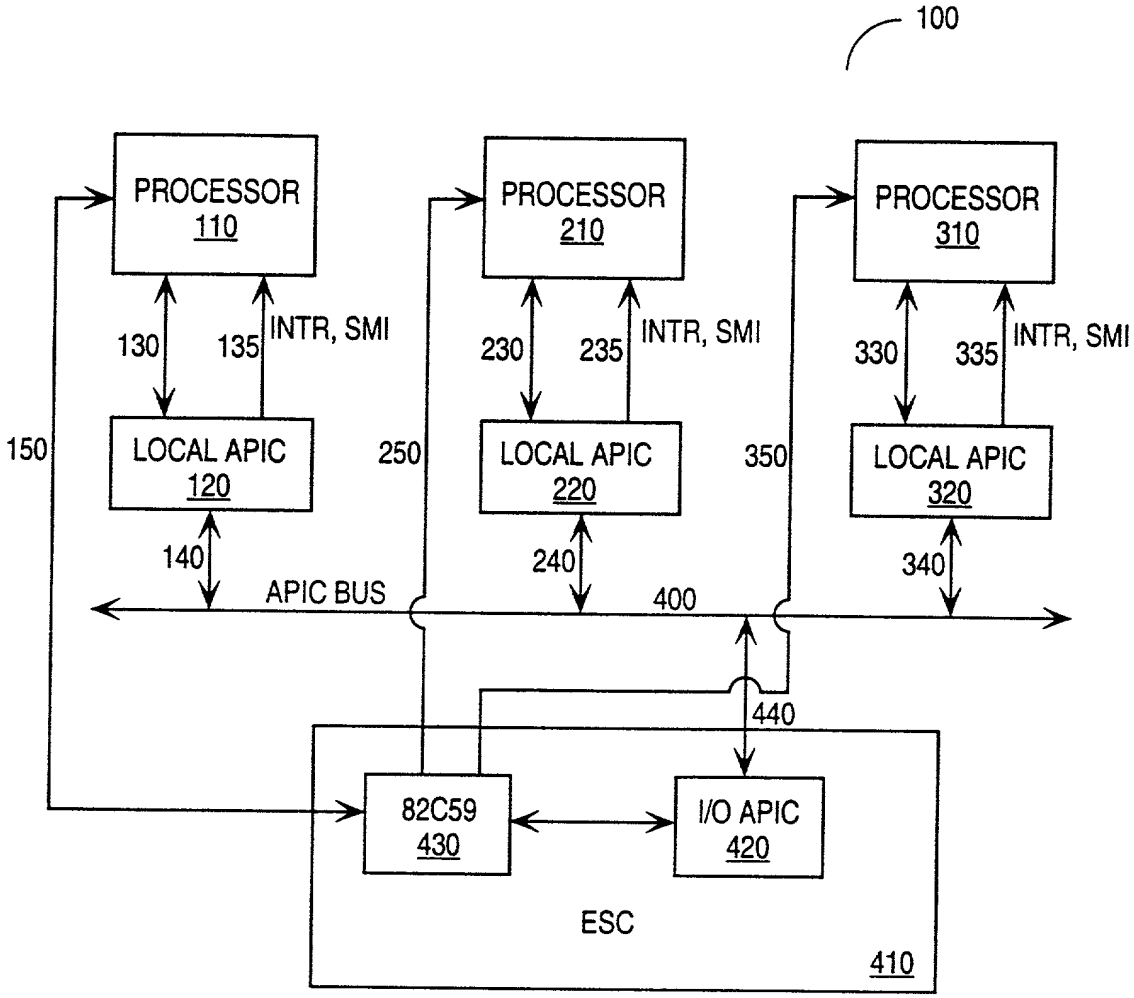
100
PROCESSOR
110
PROCESSOR
210
PROCESSOR
310
130
135
INTR, SMI
230
235
INTR, SMI
330
335
INTR, SMI
150
LOCAL APIC
120
250
LOCAL APIC
220
350
LOCAL APIC
320
140
240
340
APIC BUS
400
440
82C59
430
I/O APIC
420
ESC
410

FIG. 1

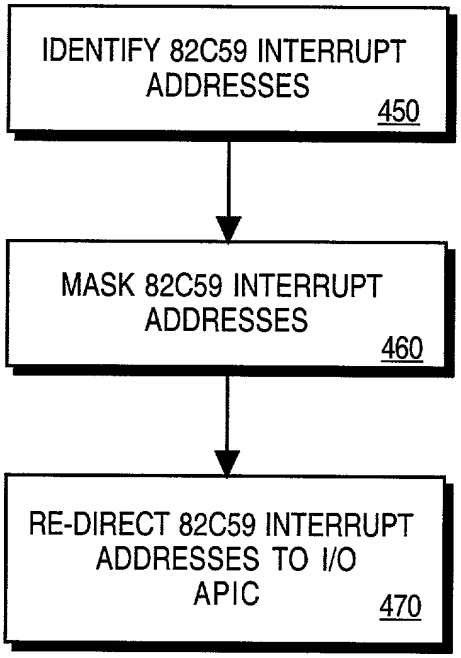
IDENTIFY 82C59 INTERRUPT ADDRESSES
450
MASK 82C59 INTERRUPT ADDRESSES
460
RE-DIRECT 82C59 INTERRUPT ADDRESSES TO I/O APIC
470

FIG. 2

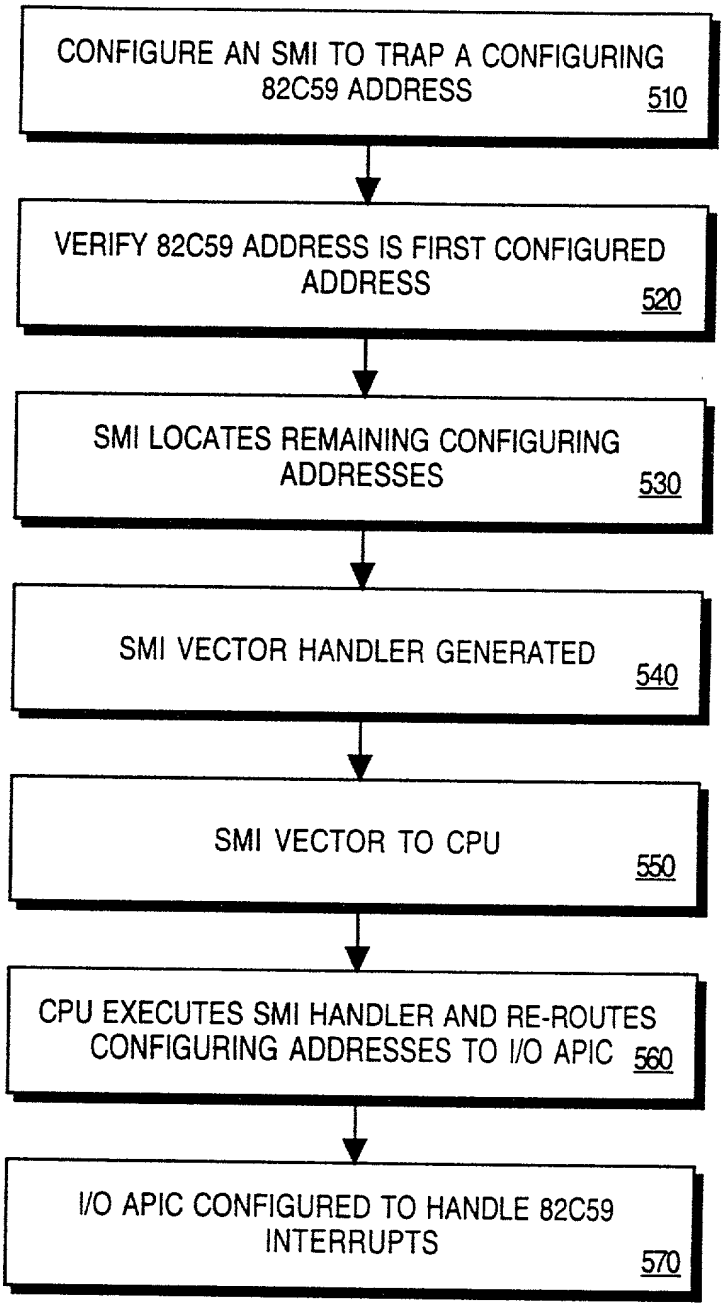
CONFIGURE AN SMI TO TRAP A CONFIGURING 82C59 ADDRESS 510
VERIFY 82C59 ADDRESS IS FIRST CONFIGURED ADDRESS 520
SMI LOCATES REMAINING CONFIGURING ADDRESSES 530
SMI VECTOR HANDLER GENERATED 540
SMI VECTOR TO CPU 550
CPU EXECUTES SMI HANDLER AND RE-ROUTES CONFIGURING ADDRESSES TO I/O APIC 560
I/O APIC CONFIGURED TO HANDLE 82C59 INTERRUPTS 570

FIG. 3